

GEOMETRY-DRIVEN IMAGE SYNTHESIS

RENDERING

BACKGROUND OF THE INVENTION

The present invention relates to computer
5 generated graphics, in particular, the modeling and
rendering of photorealistic graphics such as facial
expressions using a computer.

Computer graphics are used in many
different applications including computer games,
10 movies and web pages. With the capability of more
powerful computers, photorealistic graphics are
becoming more desired in order to provide a more
realistic experience to the computer user.

One particular area of focus has been in
15 area of synthesized photorealistic expressions of a
human face. One known technique includes "expression
mapping" (also called performance driven animation),
which has been a popular method to generate facial
animations. Using this method, a performer is located
20 in front of a computer that monitors selected points
("feature points") of the performer's face. Motions
of the feature points are then used to drive the
feature point motions of a different person's
synthesized face using the computer. However, one
25 shortcoming of this method is that the method does
not produce expression details such as wrinkles
caused by skin deformation in the synthesized face.
Thus, although the synthesized face includes, for

example, eye and mouth movements for various expressions, the synthesized face lacks photorealistic qualities because the corresponding wrinkles, folds, dimples and the like present in the skin for instance in the person's forehead, cheeks,
5 chin, etc. are not consistent with the person's overall expression.

Accordingly, a systematic method for rendering photorealistic facial expressions that
10 include appropriate changes in the skin for a given expression would be very beneficial. Aspects of such a method would be useful in other rendering applications as well.

SUMMARY OF THE INVENTION

15 A method and system uses geometry-driven feature point analysis to synthesize images including for example facial expressions. Given the feature point positions (geometry) of an expression, the method automatically synthesizes the corresponding
20 expression image, which has photorealistic and natural looking expression details.

In some applications, the number of feature points required by the synthesis system is in general more than what is available, a technique is provided
25 to infer the feature point motions from a subset by using an example-based approach. This technique can be used in an expression mapping system that monitors feature points on a user and translates the user's expression to an image rendered on a computer.

Another application of the synthesis method is on expression editing where a user indicates new locations for one or more feature points, while the system interactively generates facial expressions
5 with skin deformation details.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an exemplary computing environment for practicing the present invention.

10 FIG. 2 is a block diagram of an image processor for processing example images.

FIG. 3 is a block diagram of an image synthesizer for synthesizing images.

15 FIG. 4 is a flow chart of a method for processing example images.

FIG. 5 is a pictorial representation of feature points.

FIG. 6a is a pictorial representation of a standard or reference image.

20 FIG. 6b is a pictorial representation of blending regions.

FIG. 7 is a pictorial representation of subregions forming a complete image.

25 FIG. 8 is a flow chart of a method for synthesizing images.

FIG. 9 pictorial representations of exemplary three-dimensional synthesized images.

FIG. 10 is a flow chart for performing expression mapping.

FIG. 11 is an exemplary interface for performing expression editing.

FIG. 12 is a flow chart for performing expression editing.

5 DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

 Prior to discussing the present invention in greater detail, an embodiment of an illustrative environment in which the present invention can be used will be discussed. FIG. 1 illustrates an example
10 of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or
15 functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

20 The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the
25 invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe

computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the
5 general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or
10 implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing
15 environment, program modules may be located in both local and remote computer storage media including memory storage devices. Tasks performed by the programs and modules are described below and with the aid of figures. Those skilled in the art can
20 implement the description and figures as processor executable instructions, which can be written on any form of a computer readable media.

With reference to FIG. 1, an exemplary system for implementing the invention includes a
25 general-purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the

system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage

devices, or any other medium which can be used to store the desired information and which can be accessed by computer 100.

Communication media typically embodies
5 computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a
10 signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired
15 connection, and wireless media such as acoustic, FR, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The system memory 130 includes computer
20 storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between
25 elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of

example, and not limitation, FIG. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

The computer 110 may also include other
5 removable/non-removable volatile/nonvolatile computer storage media. By way of example only, FIG. 1 illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes
10 to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage
15 media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is
20 typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

25 The drives and their associated computer storage media discussed above and illustrated in FIG. 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In FIG. 1, for example, hard disk

drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating
5 system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are
10 different copies.

A user may enter commands and information into the computer 110 through input devices such as a keyboard 162, a microphone 163, and a pointing device 161, such as a mouse, trackball or touch pad. Other
15 input devices (not shown) may include a joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but
20 may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video
25 interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 190.

The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a hand-held device, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110. The logical connections depicted in FIG. 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 1 illustrates remote application programs 185 as residing on remote computer 180. It

will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

5

OVERVIEW

One aspect of the present invention is a computer implemented method for rendering a synthesized image that includes generating a geometric component corresponding to a selected image
10 based on identified feature points from a set of example images having the identified feature points; and generating the selected image from a composite of the set of example images based on the geometric component. The exemplary application discussed below
15 for this aspect as well as other aspects of the invention is directed to synthesis of a facial expression of a person. Nevertheless, aspects of the present invention are not intended to be limited to this application and that synthesis of any form of
20 image can benefit from aspects of the present invention including representations of other life forms or non-life forms both realistic and imaginary. Furthermore, "expressions" used herein should not be limited to that of facial expressions, but is to
25 include other forms of expression such as body expressions as well as simply movements associated with feature points for images in general.

Referring now to a facial expression by way of example, given the feature point positions of a

facial expression, to compute the corresponding expression image, one possibility would be to use some mechanism such as physical simulation to figure out the geometric deformations for each point on the face, and then render the resulting surface. The problem is that it is difficult to model the detailed skin deformations such as the expression wrinkles, and it is also difficult to render a face model so that it looks photorealistic. One aspect of the present invention is to use set of examples having the feature points and derive from the examples a desired photorealistic image expression having the appropriate wrinkles, folds, dimples, etc.

Given a set of example expressions one can generate photorealistic facial expressions through convex combination. Let $E_i = (G_i, I_i)$, $i = 0, \dots, m$, be the example expressions where G_i represents the geometry and I_i is the texture image. We assume that all the texture images I_i are pixel aligned. Let $H(E_0, E_1, \dots, E_m)$ be the set of all possible convex combinations of these examples. Then

$$H(E_0, E_1, \dots, E_m) = \left\{ \left(\sum_{i=0}^m c_i G_i, \sum_{i=0}^m c_i I_i \right) \mid \left\{ \sum_{i=0}^m c_i = 1, c_i \geq 0, i = 0, \dots, m \right\} \right\} \quad (1)$$

Pighin et al. in "Synthesizing realistic facial expressions from photographs", *Computer Graphics, Annual Conference Series*, pages 75-84, July 1998 demonstrated this ability and also developed a set of tools so that a user can use it to

interactively specify the coefficients c_i to generate the desired expressions.

Notice that each expression in the $H(E_0, E_1, \dots, E_m)$ has a geometric component $G = \sum_{i=0}^m c_i G_i$ and a
 5 texture component $I = \sum_{i=0}^m c_i I_i$.

Since the geometric component is much easier to obtain than the texture component, one aspect of the present invention uses the geometric component to infer the texture component. In
 10 particular, this method includes given the geometric component G , the geometric component G can be projected to the convex hull spanned by G_0, \dots, G_m . The resulting coefficients are then used to form a composite from the example images to obtain the
 15 desired texture image. It should be noted that this technique can be used on many different types of images and is not limited to facial expressions. Facial expressions are used herein as an exemplary application, which by itself is unique and very
 20 beneficial, but this example should not be considered limiting.

One problem with this approach is that the space of $H(E_0, E_1, \dots, E_m)$ can be limited. In the case of facial expression, a person can have expression
 25 wrinkles in different face regions, and the combinatorics is very high. In a further embodiment, the image to be synthesized is subdivided into a number of subregions. For each subregion, a geometric

component is used and is associated with this subregion to compute the desired subregion texture image. The subregions texture images are then combined, and in a further embodiment blended, to
5 produce the final image.

One potential alternative to the convex combination is to simply use the linear space without adding constraints on the coefficients c_i 's. The problem is that the coefficients resulted from the
10 linear space approximation of the geometries may contain negative coefficients as well as coefficients which are larger than 1. This can cause artifacts in the composite image.

SYSTEM OVERVIEW

15 FIGS. 2 and 3 are block diagrams illustrating an image processor 200 and an image synthesizer 300 comprising different aspects of the present invention. Referring to FIG. 2, generally, the image processor 200 receives example images 202
20 and processes the images to provide a set of registered or representative images that can used during synthesis. Typically, example images 202 are processed offline only once. Details regarding image processing are discussed below.

25 PROCESSING OF EXAMPLE IMAGES

FIG. 3 illustrates the image synthesizer 300. At run time, the synthesizer 300 receives as an input 302 the feature point positions of a desired new image such as a facial expression, accesses

registered images 204 and produces a final image 304. Details regarding image synthesis are also discussed below.

A method for image processing to generate
5 registered or representative images is illustrated in FIG. 4 at 400. At step 402, feature points are identified on each of the example images 202. The feature points denote portions of the image that will be used during synthesis. The feature points may or
10 may not include the subtle details that enable a photorealistic-synthesized image.

FIG. 5 shows a picture with feature points
500 used facial image synthesis. In FIG. 5 also are the feature points 502 of the teeth area when the
15 mouth is open. In the illustrative embodiment, there are 134 feature points in total. It should be noted it is possible to automatically compute or locate points on images such as face images as is known. However, if the number of example images 202 is
20 small, identification of the feature points in each of the images can be done manually.

Typically, after the feature points 500,
502 have been identified, the example images 202 are aligned with a standard or reference image 600 at
25 step 404, which is shown in FIG. 6A. In an application such as face synthesis, a reference image is helpful in order that the texture for the teeth can be obtained when the mouth is open. Alignment can be done by using a simple triangulation based image

warping, although more advanced techniques such as described in "Feature-based Image Metamorphosis," by T. Beier and S. Neely, in *Computer Graphics*, pages 35-42, Siggraph, July 1992, or "Animating Images with Drawings," by P. Litwinowicz and L. Williams in *Computer Graphics*, pages 235-242, Siggraph, August 1990, may be used to obtain better image quality.

As indicated above, depending on the image to be synthesized, it may be advantageous to divide the image into a number of subregions. Step 406 illustrates subdividing of the images. In the illustrative embodiment of a face, FIG. 7 illustrates exemplary subregions 700, which includes a subregion for the teeth when the mouth is open. A general guideline for subdividing the image into regions is the subregions may be small; however, details to be synthesized such as expression wrinkles should not cross the subregion boundaries. Each of the image examples 202 could be divided into subregions; however, since each of the image example 202 have been aligned with the standard image 600 and there exists a known relationship between these images, only the standard image needs to be subdivided. An image mask can be created and stored at step 406 in order to store the subdivision information where, for example, for each pixel, its subregion index is stored in its color channel.

IMAGE SYNTHESIS

FIG. 8 illustrates a method 800 for image synthesis using the set of registered or representative 204. At step 802 feature point positions for the image to be synthesized are obtained. An example of how such feature point positions can be calculated is provided below, but for purposes of synthesis, it can be assumed these are known and obtainable at step 802. Step 802 can include translation or warping of the desired feature point positions so as to be aligned with the standard or reference image 600.

At step 804, a geometric component is calculated from which a texture component will be inferred. The geometric component can be calculated as follows.

Let n denote the number of feature points. For each example expression E_i , G_i is used to denote the $2n$ dimensional vector, which includes all of the feature point positions. Let G be the feature point positions of a new expression. For each subregion R , G_i^R is used to denote the feature points of E_i which are in or at the boundary of R . Similarly G^R is used to denote the feature points of G associated with R . Given G^R , this geometric component is projected into the convex hull of G_0^R, \dots, G_m^R . In other words, the closest point in the convex hull is desired. This task can be formulated as an optimization problem:

$$\text{Minimize: } (G^R - \sum_{i=0}^m c_i G_i^R)^T (G^R - \sum_{i=0}^m c_i G_i^R) \quad (2)$$

$$\begin{aligned} \text{Subject to: } & \sum_{i=0}^m c_i = 1 \\ & c_i \geq 0, i = 0, 1, \dots, m \end{aligned}$$

Denote

$$5 \quad G = (G_0^R, G_1^R, \dots, G_m^R) \quad (3)$$

and

$$C = (c_0, c_1, \dots, c_m)^T \quad (4)$$

Then the objective function becomes

$$10 \quad C^T G^T G C - 2 G^{RT} G C + G^{RT} G^R \quad (5)$$

This is a quadratic programming formulation where the objective function is a positive semi definite quadratic form and the constraints are linear. Since G_i^R 's are in general linearly independent, the objective function is in general positive definite.

There are many known ways to solve a quadratic programming problem, for example, as described by D. G. Luenberger in *Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, 1984, or Y. Ye. *Interior Point Algorithms: Theory and Analysis*, John Wiley, 1997. In the past decade, a lot of progress has been made on the interior-point methods both in theory and in practice. Interior-point methods have become very popular for solving many practical quadratic programming problems. Using an interior point method, this approach includes iterating in the interior of the domain, which is constrained by the inequality constraints. At each

iteration, it uses an extension of Newton's method to find the next feasible point, which is closer to the optimum. Compared to the traditional approaches, interior point methods have faster convergence rate
 5 both theoretically and in practice, and they are numerically stable. Even though an interior point method usually does not produce the optimal solution (since it is an interior point), the solution is in general very close to the optimum.

10 After obtaining the coefficients c_i 's, the subregion image I^R is computed by compositing the example images together at step 806, which can be represented as:

$$I^R = \sum_{i=0}^m c_i I_i^R \quad (6)$$

15 It should be noted that if the example images have already been aligned, this step can simply be pixel-wise color blending.

 At step 808, the subregions of the image are combined to form the final synthesized image. In
 20 a further embodiment, step 808 can include blending along at least some of the subregion boundaries. Blending can be advantageous because it can avoid or minimize image discontinuity along the subregion boundaries. Blending can take many forms. In one
 25 embodiment, a fade-in-fade-out blending technique is used along the subregion boundaries. In one implementation, a weight map is used to facilitate the blending. FIG. 6B pictorially illustrates a

weight map 602, which is aligned with the standard image 600 of FIG. 6A. The thick black curves are the blending regions along the subregion boundaries. Using color channels, the intensity of the R-channel
 5 stores the blending weight, while the G-channel and the B-channel store the indexes of the two neighboring subregions, respectively.

Given a pixel in the blending region, let r denote the value of R-channel, and let i_1 and i_2 be
 10 the indexes of the two subregions. Then its blended intensity is

$$I = \frac{r}{255} * I^{i_1} + (1 - \frac{r}{255}) * I^{i_2} \quad (7)$$

It should be noted that blending may be selected based on the types of adjoining subregions.
 15 For instance, blending may not be desired along some of the boundaries where there is a natural color discontinuity such as the boundary of the eyes and the outer boundary of the lips.

After the blending step, if performed, the
 20 resulting image obtained is aligned with the standard image. In order to obtain an image that has feature point positions that are in accordance with the input feature point positions, the image is warped or translated at step 810 to obtain the final image.

25 If the final image is to include a subregion that is quite orthogonal or distinct to the other regions of the image such as the teeth region as compared to the rest of the face, a separate set

of examples images for the teeth region can be used and processed separately. In some embodiments, a smaller set of example images for such a region can be used. For instance, in facial expressions where
 5 the focus is not on speech animations that may require a lot of variations on mouth shapes such a technique can be used. Nevertheless, larger sets of example images can also be used to provide enough granularity to produce realistic animation of mouth
 10 movements or other forms of image movements when synthesized pictures comprise frames and are rendered sequentially.

At this point it should be also noted method 800 can be extended to three dimension "3D"
 15 and is not limited to two-dimensional images. In a 3D application, the feature points are not points in a substantially two-dimensional plane, but rather are positions in three dimensions. Accordingly, the synthesized images are not two-dimensional images
 20 such as facial expressions discussed above, but are synthesized 3D meshes with or without texture maps. Subregions in three dimensions can be used. To compute the sub-region blending coefficients, equation 3 is used in the same way as before except
 25 that G and G_i are $3n$ dimensional vectors. As with the two-dimensional case, a quadratic programming problem exists, which can be solved in a similar fashion such as with the same interior point method. The sub-region mesh compositing and blending along sub-region

boundaries are similar to the 2D case except that the 3D vertex positions are blended instead of the images. FIG. 9 shows examples of synthesized 3D expression images.

5 INFERRING FEATURE POINT MOTIONS FROM A SUBSET

 In practice, it may be difficult to obtain all the feature points in an image such as the facial image of FIG. 4. For example, most of the algorithms to track face features only track a limited number of
10 features along the eye brows, eyes, mouths, and noses. In an embodiment of expression mapping using aspects of the present invention discussed below, only 40 feature points are extracted from the performer. Likewise, for an application of expression
15 editing that will also be discussed below, each time when a user moves a feature point, the mostly likely movement for the rest of the feature points is ascertained.

 The following provides a method for
20 inferring or ascertaining the motions for all the feature points from a subset of feature points. The method utilizes an example-based approach. The basic idea is to learn how the rest of the feature points move from the examples. In order to have a fine-grain
25 control, which can be particularly important if only the motions of a very small number of feature points are available such as in expression editing, the feature points of an image such as a face are organized into hierarchies and hierarchical principal

component analysis on the example expressions is performed. As in the foregoing, a exemplary application will be described with respect to feature points identified on an image of a face. As
5 appreciated by those skilled in the art, this aspect can be applied to a wide variety of two and three dimensional images or representations.

In this example, three hierarchical sets of feature points are defined. At hierarchy 0, a single
10 feature point set is defined, which controls the global movement of the entire face. There are four feature point sets at hierarchy 1, each controlling the local movement of facial feature regions (left eye region, right eye region, nose region, and mouth
15 region). Each feature point set at hierarchy 2 controls details of the face regions, such as eyelid shape, lip line shape, etc. There are 16 feature point sets at hierarchy 2. Some feature points belong to several sets at different hierarchies, and they
20 are used as bridges between global and local movement of the image, herein a face, so that vertex movements from one hierarchy to another can be propagated.

For each feature point set, the displacement of all the vertices belonging to this
25 feature set for each example expression are computed. Principal component analysis on the vertex displacement vectors corresponding to the example expressions is then performed, and a lower dimensional vector space is generated. As is well-

known, principal component analysis (PCA) is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The objective of principal component analysis is to reduce the dimensionality (number of variables) of the dataset but retain most of the original variability in the data. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

The hierarchical principal component analysis result (i.e., principal components) is used to propagate vertex motions so that from the movement of a subset of feature points, the most reasonable movement for the rest of the feature points can be inferred. The basic idea is to learn from example images how the rest of the feature points move when a subset (at least one) of the vertices move.

Let v_1, v_2, \dots, v_n denote all the feature points on the image, herein a face. Let δV denote the displacement vector of all the feature points. For any given δV and a feature point set F (the set of indexes of the feature points belonging to this feature point set), $\delta V(F)$ is used to denote the sub-vector of those vertices that belong to F . Let $Proj(\delta V, F)$ denote the projection of $\delta V(F)$ into the subspace spanned by the principal components

corresponding to F . In other words, $Proj(\delta V, F)$ is the best approximation of δV (F) in the expression subspace. Given δV and $Proj(\delta V, F)$, δV is updated by $Proj(\delta V, F)$ if for each vertex that belongs to F , its
 5 displacement in δV is replaced with its corresponding value in $Proj(\delta V, F)$.

First what will be described is how to infer the motions of all the feature points from a single vertex motion. Assume vertex v_i has a motion
 10 and a vector δV is obtained where δv_i is equal to the displacement for vertex v_i , while the rest of the vertex displacements are 0. To propagate the vertex motion, the feature point set, F^* , which has the lowest hierarchy among all the feature point sets
 15 containing v_i is located. The method proceeds as follows where for each feature point set F , the flag $hasBeenProcessed(F)$ is used to denote whether F has been processed or not. Initially, $hasBeenProcessed(F)$ is set to be false for all the F .

```

20   MotionPropagation( $F^*$ )
      Begin
          Set  $h$  to be the hierarchy of  $F^*$ .
          If  $hasBeenProcessed(F^*)$  is true, return.
          Compute  $Proj(\delta V, F^*)$ .
          25   Update  $\delta V$  with  $Proj(\delta V, F^*)$ .
              Set  $hasBeenProcessed(F^*)$  to be true.
              For each feature set  $F$  belonging to
                hierarchy
                   $H - 1$  such that  $F \cap F^* \neq \emptyset$ 

```

MotionPropagation(F)

For each feature set F belonging to hierarchy

$h+1$ such that $F \cap F^* \neq \emptyset$

5

MotionPropagation(F)

End

Similarly, the motions of all the feature points can be inferred from a subset. Assume a subset of the feature points: $v_{i1}, v_{i2}, \dots, v_{ik}$ have motions.

10 The vector δV is set so that δv_{ij} is equal to the displacement vector for vertex v_{ij} for $j = 1, \dots, k$. For each vertex v_{ij} , the feature point set, F_j , is ascertained which has the lowest hierarchy among all the feature point sets containing v_{ij} , and run
15 *MotionPropagation(F_j)* (notice that now δV contains the displacement for all $v_{ij}, j = 1, \dots, k$).

ENHANCED EXPRESSION MAPPING

Expression mapping technique (also called performance-driven animation) is a simple and widely
20 used technique for facial animations. It works by computing the difference vector of the feature point positions between the neutral face and the expression face of a performer, and then adding the difference vector to the new character's face geometry. One main
25 drawback is that the resulting facial expressions may not look convincing due to the lack of expression details.

Using aspects of the present invention, a solution to this problem is provided where example images for the new character can be obtained. For instance, the example images may be obtained offline
5 through capturing or designed by an artist, where the method of FIG. 4 is used for processing. At run-time for synthesis as illustrated in FIG. 10 in method 1000, at step 1002 a geometric difference vector is calculated based on feature points of neutral face
10 and expression face of the performer. The geometric difference vector is used to obtain the desired geometry for the new character at step 1004 as in the traditional expression mapping system. Because of the difficulty of face tracking, the number of available
15 feature points is in general much smaller than the number of feature points needed by the synthesis system. So the technique described above is used to infer the motions for all the feature points used by the synthesis system at step 1006. The synthesis
20 technique of method 800 described above is then used at step 1008 to generate the texture image based on the geometry. The final results are more convincing and realistic facial expressions are obtained.

For clarification purpose, it should be
25 noted that to map a performer's expressions to the new character, example expressions from the performer are not needed. Only the feature points of the performer's expressions are needed. This is very different from the expression mapping of the prior

art, which needs example expressions for both the performer and the new character and requires the correspondence between the two sets of example expressions.

5 EXPRESSION EDITING

Another interesting application of aspects of the present invention is on interactive expression editing system. One common approach to designing facial expressions is to allow a user to interactively modify control point positions or muscle forces. The images are then warped accordingly. Aspects of the present invention can be used to enhance such systems to generate expression details interactively.

15 In an embodiment of a system including aspects of the present invention, a user is allowed to drag a feature point such as in a face, and the system interactively displays the resulting image with expression details. FIG. 11 is a snapshot of the expression editing interface where dots 1102 are the
20 feature points which the user can click on and drag, or otherwise move.

FIG. 12 illustrates a method 1200 for expression editing. At step 1202, a user drags or otherwise selects and moves a feature point. A geometry generator infers the "most likely" positions for all the feature points by using the method described above at step 1204. For example, if a user drags the feature point on the top of the nose, the

entire nose region will move instead of just this single point. With the position of the feature points ascertained, the new image can be synthesized. In one embodiment, 30-40 example expressions for the feature
5 point inference in both the expression editing and expression mapping applications are used.

When rendering the change of expression in the expression editor, a progression of a change in expression can be rendered. In one embodiment, 2-4
10 frames per second on a 2 GHz PC can be generated. Because the frame rate is not high enough, synthesis is not performed until the mouse stops moving. When the mouse stops moving, a plurality, e.g. five, geometric components for the frames between the
15 previous mouse stop and the current mouse stop are calculated, and a synthesized expression image for each frame is then rendered in the large window 1105. At the same time, we update the image in the small window. The main computation cost is the image
20 compositing. Currently the image compositing is done in software, and for every pixel the compositing operation is performed for all the example images even though some of the example images have coefficients close to 0. One way to increase the
25 frame rate is to not composite those example images whose coefficients are close to 0. Another way is to use hardware acceleration.

In summary, a geometry-driven synthesis system has been described along with a feature point

inference technique that can be used in two and three dimensional images. Each of these aspects are beneficial; however, the combination of these two techniques can be used to enhance the traditional
5 expression mapping to generate facial expression details. This is the first expression mapping system, which is capable of generating expression details while only requiring the feature point motions from the performer. In addition, an expression editing
10 application can be used where the user, while manipulating the geometric positions of the feature points, can see the resulting realistic looking facial expressions interactively. Another possibility is to extend aspects of the present invention to
15 synthesize expressions with various poses from examples. An input can be obtained for the pose parameters as well as the feature point motions, the corresponding expression from the examples would then be synthesized. Another area which aspects of the
20 present invention could be used is to handle lip motions during speech. One of the final goals is to be able to take the minimum information, such as the feature points, poses, and phonemes, of the performer and automatically synthesize the photorealistic
25 facial animations for the target character.

Although the present invention has been described with reference to particular embodiments, workers skilled in the art will recognize that

changes may be made in form and detail without departing from the spirit and scope of the invention.